

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-125548

(43)公開日 平成8年(1996)5月17日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
H 0 3 M 13/00		8730-5K		
G 1 1 B 20/18	5 3 6 B	8940-5D		
	5 4 2 C	8940-5D		
H 0 4 L 1/00		F		

審査請求 未請求 請求項の数15 O L (全 22 頁)

(21)出願番号 特願平6-262115

(22)出願日 平成6年(1994)10月26日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 中瀬 純子

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 梅本 益雄

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 瀬戸山 徹

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

(74)代理人 弁理士 秋田 収喜

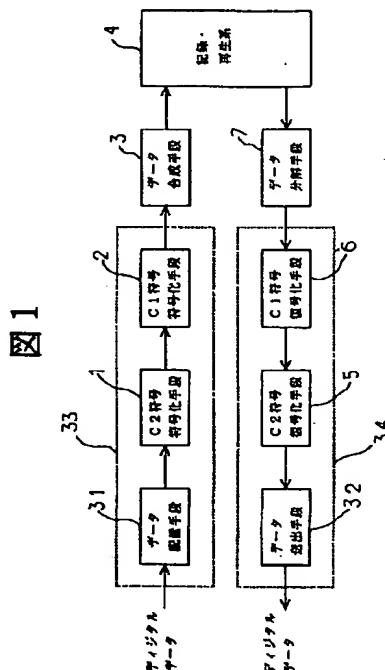
(54)【発明の名称】 誤り訂正符号化回路、誤り訂正復号化回路、誤り訂正符号化・復号化回路、および、それを用いたデジタル装置

(57)【要約】

【目的】 C1符号の符号長よりも短い長さのバースト誤りに対して効率良く誤り訂正を行うことができる誤り訂正符号化回路・誤り訂正復号化回路を提供すること。

【構成】 誤り訂正符号化回路において、積符号を形成した後に、積符号に含まれる複数のC1符号（またはC2符号）の符号語のうちN個（Nは2以上の整数）の符号語の、K個（ $1 \leq K < n1$ ）の連続するシンボルを一単位としてデータ並べ替えを行う。また、誤り訂正復号化回路において、積符号の誤り訂正復号を行う前に、

($n1 \times n2$)のシンボルのデータブロックのn1行のうちのN個（Nは2以上の整数）のデータ列の、K個（ $1 \leq K < n1$ ）の連続するシンボルを一単位としてデータの並べ替えを行い、元の積符号のデータ配置に再生する。



1

【特許請求の範囲】

【請求項1】 ($k1 \times k2$) の2次元配列に配置された情報データについて、各列毎の $k2$ 個の情報シンボル毎に ($n2, k2$) (ただし、 $n2$ は符号長) のC2符号による誤り訂正符号化を行うとともに、各行毎の $k1$ 個の情報シンボル毎に ($n1, k1$) (ただし、 $n1$ は符号長) のC1符号による誤り訂正符号化を行い、($n1 \times n2$) のシンボルから構成される積符号を生成する積符号生成手段と、

前記 ($n1 \times n2$) のシンボルの積符号に含まれる複数のC1符号の符号語のうちN個 (Nは2以上の整数) の符号語の、K個 ($1 \leq K < n1$) の連続するシンボルを一単位としてデータ並べ替えを行い、 $n1$ シンボルから構成される新たなデータ列をN個形成するデータ並べ替え手段とを具備することを特徴とする誤り訂正符号化回路。

【請求項2】 前記N個の符号語が、少なくとも二つ以上の異なる積符号に含まれることを特徴とする請求項1に記載された誤り訂正符号化回路。

【請求項3】 前記新たなデータ列のデータ並び替えに関する情報を付加するID情報付加手段を、さらに具備することを特徴とする請求項1または請求項2に記載された誤り訂正符号化回路。

【請求項4】 前記データ並べ替え手段が、複数のメモリと、前記複数のメモリの書き込み・読み出しを制御する制御部とを少なくとも具備し、前記制御部の制御に基づき前記 ($n1 \times n2$) のシンボルの積符号のC1符号の符号語のうちN個 (Nは2以上の整数) の符号語の、K個 ($1 \leq K < n1$) の連続するシンボルをデータの並べ替え順に複数のメモリに書き込み、前記複数のメモリからデータを読み出すことにより、 $n1$ シンボルから構成される新たなデータ列をN個形成することを特徴とする請求項1ないし請求項3のいずれか1項に記載された誤り訂正符号化回路。

【請求項5】 請求項1ないし請求項4のいずれか1項に記載された誤り訂正符号化回路により符号化された ($n1 \times n2$) のシンボルから構成されるデータブロックの $n1$ 行のうちのN個 (Nは2以上の整数) のデータ列の、K個 ($1 \leq K < n1$) の連続するシンボルを一単位としてデータ並べ替えを行い、元の ($n1 \times n2$) のシンボルから構成される積符号を再生するデータ並べ替え手段と、

前記再生された ($n1 \times n2$) のシンボルの積符号の、 $k2$ 個の各行毎にC1符号の誤り訂正復号を行うとともに、 $k1$ 個の各列毎にC2符号の誤り訂正復号を行う積符号復号手段とを具備することを特徴とする誤り訂正復号化回路。

【請求項6】 前記データ並べ替え手段が、複数のメモリと、前記複数のメモリの書き込み・読み出しを制御する制御部とを少なくとも具備し、前記制御部の制御の基

2

づき前記 ($n1 \times n2$) のシンボルから構成されるデータブロックの $n1$ 行のうちのN個 (Nは2以上の整数) のデータ列の、K個 ($1 \leq K < n1$) の連続するシンボルを一単位としてデータの並べ替え順に複数のメモリに書き込み、前記複数のメモリからデータを読み出すことにより、元の ($n1 \times n2$) シンボルの積符号を再生することを特徴とする請求項5に記載された誤り訂正復号化回路。

【請求項7】 ($k1 \times k2$) の2次元配列に配置された情報データについて、各列毎の $k2$ 個の情報シンボル毎に ($n2, k2$) (ただし、 $n2$ は符号長) のC2符号による誤り訂正符号化を行うとともに、各行毎の $k1$ 個の情報シンボル毎に ($n1, k1$) (ただし、 $n1$ は符号長) のC1符号による誤り訂正符号化を行い、($n1 \times n2$) のシンボルから構成される積符号を生成する積符号生成手段と、

前記 ($n1 \times n2$) のシンボルの積符号に含まれる複数のC1符号の符号語のうちN個 (Nは2以上の整数) の符号語の、K個 ($1 \leq K < n1$) の連続するシンボルを一単位としてデータ並べ替えを行い、 $n1$ シンボルから構成される新たなデータ列をN個形成する第1のデータ並べ替え手段と、

前記 ($n1 \times n2$) のシンボルから構成されるデータブロックの $n1$ 行のうちのN個 (Nは2以上の整数) のデータ列の、K個 ($1 \leq K < n1$) の連続するシンボルを一単位としてデータ並べ替えを行い、元の ($n1 \times n2$) のシンボルから構成される積符号を再生する第2のデータ並べ替え手段と、

前記再生された ($n1 \times n2$) のシンボルの積符号の、 $k2$ 個の各行毎にC1符号の誤り訂正復号を行うとともに、 $k1$ 個の各列毎にC2符号の誤り訂正復号を行う積符号復号手段とを具備することを特徴とする誤り訂正符号化・復号化回路。

【請求項8】 前記N個の符号語が、少なくとも二つ以上の異なる積符号に含まれることを特徴とする請求項7に記載された誤り訂正符号化・復号化回路。

【請求項9】 前記新たなデータ列のデータ並び替えに関する情報を付加するID情報付加手段を、さらに具備することを特徴とする請求項7または請求項8に記載された誤り訂正符号化・復号化回路。

【請求項10】 前記第1のデータ並べ替え手段、および、第2のデータ並べ替え手段が、複数の符号化用メモリと、複数の復号化用メモリと、前記複数の符号化用メモリと復号化用メモリの書き込み・読み出しを制御する制御部とを少なくとも具備し、前記制御部の制御に基づき前記 ($n1 \times n2$) のシンボルの積符号のC1符号の符号語のうちN個 (Nは2以上の整数) の符号語の、K個 ($1 \leq K < n1$) の連続するシンボルをデータの並べ替え順に複数の符号化用メモリに書き込み、前記複数の符号化用メモリからデータを読み出すことにより、 $n1$

シンボルから構成される新たなデータ列をN個形成し、また、前記制御部の制御に基づき前記 $(n1 \times n2)$ のシンボルから構成されるデータブロックの $n1$ 行のうちのN個(Nは2以上の整数)のデータ列の、K個($1 \leq K < n1$)の連続するシンボルを一単位としてデータの並べ替え順に複数の復号化用メモリに書き込み、前記複数の復号化用メモリからデータを読み出すことにより、元の $(n1 \times n2)$ シンボルの積符号を再生することを特徴とする請求項7ないし請求項9のいずれか1項に記載された誤り訂正符号化・復号化回路。

【請求項11】 前記複数の符号化用メモリと、前記複数の復号化用メモリとが、同一の複数のメモリから構成されることを特徴とする請求項10に記載された誤り訂正符号化・復号化回路。

【請求項12】 請求項1ないし請求項4のいずれか1項に記載された誤り訂正符号化回路、および、請求項5または請求項6に記載された誤り訂正復号化回路を具備することを特徴とするデジタル記録装置。

【請求項13】 請求項7ないし請求項11のいずれか1項に記載された誤り訂正符号化・復号化回路を具備することを特徴とするデジタル記録装置。

【請求項14】 請求項1ないし請求項4のいずれか1項に記載された誤り訂正符号化回路、および、請求項5または請求項6に記載された誤り訂正復号化回路を具備することを特徴とするデジタル通信装置。

【請求項15】 請求項7ないし請求項11のいずれか1項に記載された誤り訂正符号化・復号化回路を具備することを特徴とするデジタル通信装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、誤り訂正符号として積符号を用いる誤り訂正符号化回路、誤り訂正復号化回路、誤り訂正符号化・復号化回路、および、それを用いたデジタル通信装置又はデジタル記録装置に関する。

【0002】

【従来の技術】デジタル通信システムには、通信回線上、あるいは、無線を用いて画像、音声等のデジタルデータを伝送するデジタル通信システム、デジタル放送システム等がある。

【0003】また、デジタル記録装置には、磁気テープ装置、磁気ディスク装置、光ディスク装置等がある。

【0004】前記したようなデジタル記憶装置、あるいは、デジタル通信装置においては、データの信頼性を確保するための技術として誤り訂正技術が用いられている。

【0005】誤り訂正とは、データに生じた誤り、例えば、デジタル記録装置ならば記録したビットの'1'が'0'に、あるいは、'0'が'1'に再生される場合に、元の正しいビットに戻す信号処理である。

【0006】デジタル記録装置、または、デジタル通信装置において生じる誤りには、一般に、ランダム誤りとバースト誤りの2種類の誤りがある。

【0007】ここで、ランダム誤りは、見かけ上ほぼランダムにビット単位で生じる誤りであり、これに対して、バースト誤りとは、ある期間複数ビットにわたって連続して生じる誤りである。

【0008】このような2種類の誤りを効率的に訂正するための符号化方法の一つに、情報(データ)を二次元配列に配置し、異なる方向に対して誤り訂正符号化を複数回行う方法が公知である。

【0009】例えば、誤り訂正符号化を2回行う二重符号化の例としては、デジタルVTR等で用いられている積符号がある。

【0010】積符号とは、情報を二次元配列に配置し、縦、横2方向に二重の誤り訂正符号化を行うものである。

【0011】以下、デジタルVTRを例にとり、積符号符号化の手順とその訂正能力について簡単に説明する。

【0012】デジタルVTRでは、誤り訂正符号としてリードソロモン符号が用いられている。

【0013】リードソロモン符号は、複数ビットを1シンボルとし、シンボル単位で訂正を行う誤り訂正符号であり、例えば、業務用デジタルVTRでは、8ビットを1シンボルとするリードソロモン符号を用いている。

【0014】リードソロモン符号では、情報に付加するパリティが $2t$ シンボルのとき、誤りの位置がわからない場合には最大 t シンボル、誤りの位置が予め分かっている場合には $2t$ シンボルまで訂正できる。

【0015】誤り位置がわからない場合の訂正をランダム訂正、誤りの位置が予め分かっている場合の訂正をイレージャ訂正という。

【0016】なお、リードソロモン符号の詳細については今井秀樹著「符号理論」(電子情報通信学会編、1990)等の専門書を参照されたい。

【0017】図22は、積符号の符号化方法を説明するための図である。

【0018】以下、図22を用いて、積符号の符号化方法を説明する。

【0019】積符号の例として、C1符号が $(n1, k1)$ RS符号、C2符号が $(n2, k2)$ RS符号の例を示す。

【0020】ここで、 (n, k) RS符号とは、符号長が n で情報シンボル数が k のリードソロモン符号を示す。

【0021】まず、図22に示すように記録する情報を二次元配列($k1$ シンボル $\times k2$ シンボル)に配置する。

【0022】ここで、図22中の格子の一つが1シンボ

ルに相当する。

【0023】次に、縦方向にC2符号の誤り訂正符号化を行い($n2-k2$)シンボルのパリティを付加し、($n2, k2$)RS符号の符号語を生成する。

【0024】次に、横方向にC1符号の符号化を行い($n1-k1$)シンボルのパリティを付加し、($n1, k1$)RS符号の符号語を生成する。

【0025】前記C1符号、C2符号の2段階の符号化により得られた($n1 \times n2$)シンボルの符号が積符号である。

【0026】以後、本明細書中では($n1 \times n2$)のシンボルの2次元ブロックを指して積符号ブロックと称す。

【0027】このような符号化により得られた積符号ブロックのデータは、通常、C1符号の符号語を単位として、上段から順に図中矢印の方向に磁気テープ上に記録される。

【0028】一方、再生時には、磁気テープから連続的に再生されるデータを記録時のデータ順と同じ順で2次元配列に配置し、C1符号、C2符号の順で誤り訂正復号を行う。

【0029】C1符号では、パリティ数が($n1-k1$)シンボルであるため訂正できるシンボル数は最大($n1-k1$)/2シンボルである。

【0030】また、C1符号の誤り訂正復号で訂正できなかった符号語の全シンボルには誤り検出フラグを立てる。

【0031】C2符号の誤り訂正復号では、C1符号の復号時に付加された誤り検出フラグの数が($n2-k2$)個以下の場合に、誤り検出フラグ位置を誤りの位置としてイレージャ訂正を行う。

【0032】C2符号では、パリティ数が($n2-k2$)であるためイレージャ訂正によって最大($n2-k2$)シンボルまで訂正できる。

【0033】磁気テープ上で連続した($(n2-k2) \times n1$)シンボルの誤りは、C2符号で見ると($n2-k2$)シンボルの誤りであるため訂正できる。

【0034】即ち、前記積符号においては、磁気テープ上に生じた($(n2-k2) \times n1$)シンボルのバースト誤りを訂正できる。

【0035】なお、誤り検出フラグの数が($n2-k2$)個を超えている場合にはC1符号同様にランダム訂正を行う。

【0036】前記積符号の符号化方法において、積符号ブロックの構成は変更せずに、より誤り訂正効率を向上させる方法が従来から知られている。

【0037】前記積符号ブロックの構成は変更せずに、より誤り訂正能力を向上させる従来方法について以下簡単に説明する。

【0038】まず、N個(Nは2以上の整数)の積符号

ブロックを、図23に示すように横方向に並べ上段から順に図中矢印の方向に磁気テープにデータを記録する。

【0039】このようにすると異なる積符号ブロックのC1符号の符号語のデータが連続して記録されることになる。

【0040】これは、異なる積符号ブロック間でC1符号の符号語単位のデータ並べ替えを行っていることに等しい。

【0041】このデータ並べ替えにより磁気テープ上で連続するバースト誤りがN個の積符号ブロックに振り分けられ、各積符号ブロックで見るとバースト誤りの長さが見かけ上 $1/N$ となる。

【0042】例えば、磁気テープ上に生じた($n1 \times N$)シンボルの長さのバースト誤りは、各積符号ブロックではそれぞれ $n1$ シンボルのバースト誤りとなる。

【0043】そのため訂正できるバースト誤りの長さは、各積符号ブロックのデータをまとめて記録する場合のN倍の($(n2-k2) \times n1 \times N$)シンボルとなる。

【0044】即ち、前記の符号語単位のデータ並べ替えによって積符号の持つバースト誤りに対する訂正能力をより有効に活用できることになる。

【0045】

【発明が解決しようとする課題】前記した従来のデータ並べ替え方法では、C1符号の符号語単位でデータの並べ替えを行っており、この並べ替えは、C1符号の符号長を超える長さのバースト誤りを複数の積符号ブロックに分散させることができる。

【0046】しかしながら、符号語を一まとめにして並べ替えるため、C1符号の符号語内に生じた符号長以下の長さのバースト誤りは分散されない。

【0047】そのため、従来の方法では必ずしも十分に誤り訂正符号の持つ訂正能力を有効活用できず、目標とする誤り率を得られないことがある。

【0048】本発明は、前記従来技術の問題点を解決するためになされたものであり、本発明の目的は、誤り訂正符号化回路・誤り訂正復号化回路において、C1符号の符号長よりも短い長さのバースト誤りに対して効率良く誤り訂正を行うことができる技術を提供することにある。

【0049】本発明の前記目的並びにその他の目的及び新規な特徴は、本明細書の記載及び添付図面によって明らかにする。

【0050】

【課題を解決するための手段】本願において開示される発明のうち、代表的なものの概要を簡単に説明すれば、下記の通りである。

【0051】(1) ($k1 \times k2$)の2次元配列に配置された情報データについて、各列毎の $k2$ 個の情報シンボル毎に($n2, k2$) (ただし、 $n2$ は符号長)のC

2 符号による誤り訂正符号化を行うとともに、各行毎の $k1$ 個の情報シンボル毎に $(n1, k1)$ (ただし、 $n1$ は符号長) の $C1$ 符号による誤り訂正符号化を行い、
 $(n1 \times n2)$ のシンボルから構成される積符号を生成する積符号生成手段と、前記 $(n1 \times n2)$ のシンボルの積符号に含まれる複数の $C1$ 符号の符号語のうち N 個 (N は 2 以上の整数) の符号語の、 K 個 ($1 \leq K < n1$) の連続するシンボルを一単位としてデータ並べ替えを行い、 $n1$ シンボルから構成される新たなデータ列を N 個形成するデータ並べ替え手段とを具備することを特徴とする。

【0052】(2) 前記(1)の手段において、前記 N 個の符号語が、少なくとも二つ以上の異なる積符号に含まれることを特徴とする。

【0053】(3) 前記(1)または(2)の手段において、前記新たなデータ列のデータ並び替えに関する情報を付加するID情報付加手段を、さらに具備することを特徴とする。

【0054】(4) 前記(1)ないし(3)の手段において、前記データ並べ替え手段が、複数のメモリと、前記複数のメモリの書き込み・読み出しを制御する制御部とを少なくとも具備し、前記制御部の制御に基づき前記 $(n1 \times n2)$ のシンボルの積符号の $C1$ 符号の符号語のうち N 個 (N は 2 以上の整数) の符号語の、 K 個 ($1 \leq K < n1$) の連続するシンボルをデータの並べ替え順に複数のメモリに書き込み、前記複数のメモリからデータを読み出すことにより、 $n1$ シンボルから構成される新たなデータ列を N 個形成することを特徴とする。

【0055】(5) 前記(1)ないし(4)の手段により符号化された $(n1 \times n2)$ のシンボルから構成されるデータブロックの $n1$ 行のうちの N 個 (N は 2 以上の整数) のデータ列の、 K 個 ($1 \leq K < n1$) の連続するシンボルを一単位としてデータ並べ替えを行い、元の $(n1 \times n2)$ のシンボルから構成される積符号を再生するデータ並べ替え手段と、前記再生された $(n1 \times n2)$ のシンボルの積符号の、 $k2$ 個の各行毎に $C1$ 符号の誤り訂正復号を行うとともに、 $k1$ 個の各列毎に $C2$ 符号の誤り訂正復号を行う積符号復号手段とを具備することを特徴とする。

【0056】(6) 前記(5)の手段において、前記データ並べ替え手段が、複数のメモリと、前記複数のメモリの書き込み・読み出しを制御する制御部とを少なくとも具備し、前記制御部の制御に基づき前記 $(n1 \times n2)$ のシンボルから構成されるデータブロックの $n1$ 行のうちの N 個 (N は 2 以上の整数) のデータ列の、 K 個 ($1 \leq K < n1$) の連続するシンボルを一単位としてデータの並べ替え順に複数のメモリに書き込み、前記複数のメモリからデータを読み出すことにより、元の $(n1 \times n2)$ シンボルの積符号を再生することを特徴とする。

【0057】(7) ($k1 \times k2$) の 2 次元配列に配置された情報データについて、各列毎の $k2$ 個の情報シンボル毎に $(n2, k2)$ (ただし、 $n2$ は符号長) の $C2$ 符号による誤り訂正符号化を行うとともに、各行毎の $k1$ 個の情報シンボル毎に $(n1, k1)$ (ただし、 $n1$ は符号長) の $C1$ 符号による誤り訂正符号化を行い、
 $(n1 \times n2)$ のシンボルから構成される積符号を生成する積符号生成手段と、前記 $(n1 \times n2)$ のシンボルの積符号に含まれる複数の $C1$ 符号の符号語のうち N 個 (N は 2 以上の整数) の符号語の、 K 個 ($1 \leq K < n1$) の連続するシンボルを一単位としてデータ並べ替えを行い、 $n1$ シンボルから構成される新たなデータ列を N 個形成する第1のデータ並べ替え手段と、前記 $(n1 \times n2)$ のシンボルから構成されるデータブロックの $n1$ 行のうちの N 個 (N は 2 以上の整数) のデータ列の、 K 個 ($1 \leq K < n1$) の連続するシンボルを一単位としてデータ並べ替えを行い、元の $(n1 \times n2)$ のシンボルから構成される積符号を再生する第2のデータ並べ替え手段と、前記再生された $(n1 \times n2)$ のシンボルの積符号の、 $k2$ 個の各行毎に $C1$ 符号の誤り訂正復号を行うとともに、 $k1$ 個の各列毎に $C2$ 符号の誤り訂正復号を行う積符号復号手段とを具備することを特徴とする。

【0058】(8) 前記(7)の手段において、前記 N 個の符号語が、少なくとも二つ以上の異なる積符号に含まれることを特徴とする。

【0059】(9) 前記(7)または(8)の手段において、前記新たなデータ列のデータ並び替えに関する情報を付加するID情報付加手段を、さらに具備することを特徴とする。

【0060】(10) 前記(7)ないし(9)の手段において、前記第1のデータ並べ替え手段、および、第2のデータ並べ替え手段が、複数の符号化用メモリと、複数の復号化用メモリと、前記複数の符号化用メモリと復号化用メモリの書き込み・読み出しを制御する制御部とを少なくとも具備し、前記制御部の制御に基づき前記 $(n1 \times n2)$ のシンボルの積符号の $C1$ 符号の符号語のうち N 個 (N は 2 以上の整数) の符号語の、 K 個 ($1 \leq K < n1$) の連続するシンボルをデータの並べ替え順に複数の符号化用メモリに書き込み、前記複数の符号化用メモリからデータを読み出すことにより、 $n1$ シンボルから構成される新たなデータ列を N 個形成し、また、前記制御部の制御に基づき前記 $(n1 \times n2)$ のシンボルから構成されるデータブロックの $n1$ 行のうちの N 個 (N は 2 以上の整数) のデータ列の、 K 個 ($1 \leq K < n1$) の連続するシンボルを一単位としてデータの並べ替え順に複数の復号化用メモリに書き込み、前記複数の復号化用メモリからデータを読み出すことにより、元の $(n1 \times n2)$ シンボルの積符号を再生することを特徴とする。

【0061】(11)前記(10)の手段において、前記複数の符号化用メモリと、前記複数の復号化用メモリとが、同一の複数のメモリから構成されることを特徴とする。

【0062】

【作用】前記各手段によれば、誤り訂正符号化回路において、積符号を形成した後に、積符号に含まれる複数のC1符号(またはC2符号)の符号語のうちN個(Nは2以上の整数)の符号語の、K個($1 \leq K < n1$)の連続するシンボルを一単位としてデータ並べ替えを行う。

【0063】また、誤り訂正復号回路において、積符号の誤り訂正復号を行う前に、($n1 \times n2$)のシンボルのデータブロックのn1行のうちのN個(Nは2以上の整数)のデータ列の、K個($1 \leq K < n1$)の連続するシンボルを一単位としてデータの並べ替えを行い、元の積符号のデータ配置に再生する。

【0064】これにより、記録媒体または通信路のn1シンボルの範囲において生じたLシンボル($L < n1$)のバースト誤りを、N個の符号語にほぼ L/N シンボルずつ分散することが可能となる。

【0065】それにより、C1符号が符号語あたりtシンボルまで訂正可能な符号である場合には、n1シンボルの範囲に約Ntシンボルの量の誤りが生じても訂正することが可能となり、即ち、C1符号で訂正できる誤りの数を見かけ上約N倍にすることが可能となる。

【0066】

【実施例】以下、図面を参照して本発明の実施例を詳細に説明する。

【0067】なお、実施例を説明するための全図において、同一機能を有するものは同一符号を付け、その繰り返しの説明は省略する。

【0068】なお、本発明は、積符号を用いて誤り訂正を行うデジタル通信装置及びデジタル記録装置に共通して適用できるものであるが、以下の実施例では本発明をデジタル記録装置に適用した場合について説明する。

【0069】【実施例1】図1は、本発明の一実施例(実施例1)である誤り訂正符号化回路、および、誤り訂正復号化回路を用いたデジタル記録装置の概略構成を示す図である。

【0070】図1に示すデジタル記録装置は、コンピュータの入出力装置として使用されるデジタル記録装置である。

【0071】本実施例1の誤り訂正符号化回路は、積符号形成手段33とデータ合成手段とから構成され、また、本実施例1の誤り訂正復号化回路は、データ分解手段7と積符号復号手段33とから構成される。

【0072】本実施例1の誤り訂正符号化回路では、始めに、データ配置手段31で、コンピュータからの記録情報を2次元配列($k1$ シンボル $\times k2$ シンボル)に配

置する。

【0073】次に、C2符号符号化手段1、C1符号符号化手段2によって、C2符号、C1符号の符号化を行った後に、データ合成手段3によりシンボル単位のデータ並べ替えを行う。

【0074】その後、記録・再生系4において、本実施例1の誤り訂正符号化回路により符号化されたデータを記録媒体に記録する。

【0075】また、本実施例1の誤り訂正復号化回路では、再生時に記録・再生系4において再生されたデータの系列を、データ分解手段5によって並べ替え、元の積符号のデータの並びに再生する。

【0076】その後、C1符号復号手段6、C2符号復号手段7により積符号の復号を行い、データ送出手段32から再生情報を、コンピュータに送出する。

【0077】なお、本実施例1の誤り訂正符号化回路では、データ配置手段31で、コンピュータからの記録情報を2次元配列($k1$ シンボル $\times k2$ シンボル)に配置した後で、C2符号符号化手段1でC2符号の符号化を行うようにしたが、これに代えて、コンピュータからC2符号の1符号語分の記録情報($k1$ シンボル)が入力された時点でそれに対するC2符号の符号化を行い、 $k1$ シンボル $\times k2$ シンボルの記録情報に対するC2符号の符号化が終了した時点で $k1$ シンボル $\times n2$ シンボルの記録情報を2次元配列に配置するようにすることも可能である。

【0078】図2は、前記データ合成手段、データ分解手段の動作を説明するための図である。

【0079】次に、図2を用いて、前記データ合成手段、データ分解手段の動作について説明する。

【0080】図2は、データ合成・分解すべき(9×6)シンボルの積符号ブロックを示しており、図2において、C1符号は(9、5)RS符号、C2符号は(6、4)RS符号であるとする。

【0081】ただし、(n, k)RS符号とは、符号長nシンボル、情報シンボル数kのリードソロモン符号を示す。

【0082】C1符号では、2シンボルの誤りまで訂正可能であり、また、C2符号ではC1符号復号時の誤り検出フラグを用いる場合には2シンボルの誤り、あるいは、誤り検出フラグを用いない場合には1シンボルの誤りを訂正可能である。

【0083】図2において、積符号ブロックを構成する54個のシンボルにA1、A2、…、A54の番号を付与し、また、C1符号の符号語は上段から順に符号語1、2、…、6であるとする。

【0084】本実施例1では、連続する二つの符号語(符号語1と符号語2、符号語3と符号語4、符号語5と符号語6)を一グループとし、各グループ毎にデータ合成を行う。

【0085】データ合成手段3では、図2の合成ブロックに示すように1シンボル単位で二つの符号語のデータが互い違いになるようにデータの並べ替えを行う。

【0086】その結果、符号語と等しい長さを持つ新たなデータブロックが各グループ毎に二つ生成されることになる。

【0087】例えば、符号語1と符号語2からは(A1、A10、A2、A11、A3、A12、A4、A13、A5)、(A14、A6、A15、A7、A16、A8、A17、A9、A18)の二つのデータブロック 10 が合成される。

【0088】これら二つのデータブロックをそれぞれ合成語1、合成語2とする。

【0089】以下同様に、合成語3、4、…、6を合成する。

【0090】次に、このように合成されたデータ合成後の(9×6)シンボルのデータブロックのデータを上段から順に左から右に記録媒体に記録する(A1、A10、A2、A11、A3、A12、A4、A13、A5、A14、A6、A15、A7、A16、A8、A17、A9、A18、…の順)。 20

【0091】これに対し再生時には、データ分解手段5によりデータを並べ替え、データ合成前の積符号ブロックのデータ並びに再生する。

【0092】図3は、図2示すデータ合成及びデータ分解の効果について説明するための図である。

【0093】次に、図3を用いて、図2に示すデータ合成及びデータ分解の効果について説明する。

【0094】図3の合成ブロックは、再生時データ分解前のデータブロックを示す。 30

【0095】ここで、ハッチングが施してあるシンボルに誤りが生じているとする。

【0096】即ち、上から1段目には4シンボル、3、4段目には共に3シンボル、5段目には4シンボルの誤りがそれぞれ生じている。

【0097】比較のため、積符号ブロックに対しこのような分布の誤りが生じた場合について考える。

【0098】この場合、C1符号復号で1、3、4、5段目の誤りを訂正できず、これら4段に対し誤り検出フラグが付加される。

【0099】C2符号復号ではC1復号時に付加された誤り検出フラグの数が2を超えているためイレージャ訂正が行えずランダム訂正を行う。

【0100】この結果、C2符号の符号語で誤りが1シンボル以下である符号語については誤りを訂正できるが、その他の符号語(左から4、5、6、7列目の符号語)の誤りは訂正されずに残ることになる。

【0101】即ち、データ合成を行わない積符号ブロックに対し、図3に示す合成ブロックのような分布の誤りが生じた場合には、積符号の復号後に11シンボルの誤 50

りが残ることになる。

【0102】一方、データ合成を行った積符号ブロックに、図3に示す合成ブロックのような分布の誤りが生じた場合には、積符号の復号の前にデータ分解を行うため、誤り分布は、図3に示す積符号ブロックのような誤り分布に変換される。

【0103】図3に示す積符号ブロックに対し復号を行うと、上から4段目を除いてはC1符号の復号で誤りを訂正できる。

【0104】4段目については、誤り検出フラグを立てることによりC2符号復号時に訂正できる。

【0105】即ち、データ合成を行う場合には、図3に示す合成ブロックの全ての誤りを訂正できることになる。

【0106】したがって、図3に示す合成ブロックのような分布の誤りが生じた場合、データ合成を行ったときには行わないときに比べ11シンボル多く誤りを訂正できる。

【0107】なお、前記説明では、理解しやすくするため、データを並べ替えることによって全ての誤りを訂正することが可能な場合について説明したが、誤りの種類によっては、データを並べ替えても全ての誤りを訂正することは不可能な場合があるな、そのような場合でも、訂正能力が向上することには変わりはない。

【0108】また、本実施例1では、データ合成を、C1の二つの符号語で行う場合について説明したが、データ合成は必ずしも二つの符号語で行う必要はなく、三つ以上の符号語のデータを合成してもよい。

【0109】また、必ずしも連続する符号語を用いてデータ合成を行う必要はなく位置的に離れた符号語同士を合成してもよい。

【0110】さらに、データ合成時のデータ並べ替え方法は、必ずしも上記実施例で述べた方法に限定されるものではない。

【0111】例えば、図2の符号語1、符号語2から合成語1として(A1、A11、A3、A13、A5、A15、A7、A17、A9)、合成語2として(A10、A2、A12、A4、A14、A6、A16、A8、A18)の組み合わせを生成してもよい。

【0112】図4は、本実施例1におけるデータ合成手段・データ分解手段の回路構成の一例を示す回路図である。

【0113】図4において、8はデータ合成部、9はデータ分解部、10は制御部であり、データ合成部8は、メモリA13、メモリB14、メモリC15、メモリD16と、それら四つのメモリ出力から一つを選択して出力するマルチプレクサ11からなる。

【0114】また、データ分解部9は、メモリE17、メモリF18、メモリG19、メモリH20と、それら四つのメモリ出力のうち一つを選択して出力するマルチ

プレクサ12から成る。

【0115】前記メモリA~H(13~20)は、それぞれ少なくとも符号語一つ分のデータが格納できるデータ容量を持つ。

【0116】前記図2に示すデータ合成を例に挙げて、図4に示すデータ合成部8・データ分解部9の動作について説明する。

【0117】まず、データ合成部8の動作について説明する。

【0118】図5は、図4に示すデータ合成部8の動作を示すタイムチャートである。

【0119】また、図6は、図5に示すタイムチャートの部分拡大図である。

【0120】なお、図5、図6のタイムチャートにおいて「W」、「R」とあるのはそれぞれメモリに書き込みを行うこと、メモリからデータを読み出すことを表している。

【0121】ただし、「W'」とあるのは、図6に示すようにその期間の前半又は後半の期間のみメモリに対して書き込みを行うことを示している。

【0122】また、図6のタイムチャートにおいて、「W」の下に括弧内の数字はメモリのアドレスを示している。

【0123】ここで、図5に示すようにデータ合成部に対する入力が符号語1、2、3、…の順であったとする。

【0124】まず、符号語1を構成する(A1、A2、A3、A4、A5、A6、A7、A8、A9)のシンボルのうちA1、A2、A3、A4、A5は、メモリA13の奇数アドレスに対して順次書き込まれ、A6、A7、A8、A9は、メモリB14の偶数アドレスに書き込まれる。

【0125】次に、符号語2を構成する(A10、A11、A12、A13、A14、A15、A16、A17、A18)のシンボルのうちA10、A11、A12、A13は、メモリA13の偶数アドレスに対し書き込まれ、A14、A15、A16、A17、A18は、メモリB14の奇数アドレスに書き込まれる。

【0126】符号語2の全てのシンボルのメモリに対する書き込みが終了した時点で、メモリA13には符号語1、2の前半のシンボルから成る合成語1、メモリB14には符号語1、2の後半シンボルから成る合成語2が格納されている。

【0127】さらに、符号語3、4のシンボルについても同様にメモリC15、メモリD16に対して書き込まれ、合成語3、4が生成される。

【0128】以後、符号語5、6はメモリA13、メモリB14に、符号語7、8はメモリC15、メモリD16に、というように以後メモリA13とメモリB14、メモリC15とメモリD16の組み合わせを交互に切り

替えながら同様に書き込まれる。

【0129】一方、メモリA13、メモリB14、メモリC15、メモリD16上に生成された合成語1、2、3、4はマルチプレクサ11を通じて記録データとして読み出される。

【0130】マルチプレクサ11の制御は制御部10によって行われ、符号語3がデータ合成部8に入力されている期間、即ち合成語1が出力される期間にはメモリA13を選択し、符号語4がデータ合成部8に入力されている期間、即ち合成語2が出力される期間にはメモリB14を選択する。

【0131】次に、データ分解部9の動作について説明する。

【0132】データ分解部9では、図2に示す合成ブロックのデータ配置から図2に示す積符号ブロックのデータ配置に変換する。

【0133】図7は、図4に示すデータ分解部9の動作を示すタイムチャートである。

【0134】また、図8は、図7に示すタイムチャートの部分拡大図である。

【0135】なお、図7、図8のタイムチャートにおける「W」、「R」は、前記図5、12に示すタイムチャートと同様、それぞれメモリに対するデータの書き込み、メモリからのデータ読み出しを表している。

【0136】また、「W'」とあるのは、図8に示すようにメモリに対する書き込みの実行を1シンボルおきに行うことを示している。

【0137】ここで、データ分解部9に対して入力が合成語1、2、3、…の順である場合を考える。

【0138】まず、合成語1を構成する(A1、A10、A2、A11、A3、A12、A4、A13、A5)のシンボルのうち一つおきのA1、A2、A3、A4、A5が、メモリE17の連続するアドレスに対し順次書き込まれ、同様に、A10、A11、A12、A13がメモリF18に対し順次書き込まれる。

【0139】合成語2(A14、A6、A15、A7、A16、A8、A17、A9、A18)のシンボルのうち一つおきのA6、A7、A8、A9は、メモリE17の合成語1のデータの後に続いて書き込まれ、A14、A15、A16、A17、A18はメモリF18の合成語1のデータの後に続いて書き込まれる。

【0140】合成語2の全てのシンボルの書き込みが終了した時点で、メモリE17、メモリF18にはそれぞれ符号語1、2を構成するシンボルがデータ分解前と同じ順番で格納されている。

【0141】同様に、合成語3、4についてはメモリG19、メモリH20を用いて同様にデータ分解が行われ、符号語3、4を得る。

【0142】合成語5、6のデータ分解には、メモリE17、メモリF18、合成語7、8のデータ分解にはメ

メモリG19、メモリH20をそれぞれ用いる。

【0143】以後の合成語についても同様に、メモリE17とメモリF18、メモリG19とメモリH20の2種類の組み合わせを交互に切り換えながらデータ分解が行われる。

【0144】一方、メモリ上に形成された符号語1、2、3、4はマルチプレクサ12を通じて出力される。

【0145】前記したように、図4に示すデータ合成部・データ分解部を用いることによってデータ合成、データ分解を実行できる。

【0146】この結果、信頼性の高いデジタル記録装置が実現できる。

【0147】なお、図4では、二つの符号語のデータを合成・分解する場合のデータ合成部・データ分解部を示したが、三つ以上の符号語のデータを合成・分解するデータ合成部・分解部も同様に構成できる。

【0148】L個（Lは2以上の整数）の符号語のデータの合成・分解を行うデータ合成部・分解部の構成は、データ合成部、データ分解部のそれぞれに、符号語一つ分のデータを格納できるメモリを2L個とそれら2L個のメモリの出力のうち一つを選択するマルチプレクサの組み合わせを備え、さらに、これらデータ合成部とデータ分解部を制御する制御部を備えたものとなる。

【0149】図9は、本実施例1におけるデータ合成手段・データ分解手段の回路構成の他の例を示す回路図であり、データ合成とデータ分解に用いるメモリを共用するようにした回路図である。

【0150】図9において、21、22はマルチプレクサ、23は制御部であり、マルチプレクサ21は、C1符号符号器の出力と再生データのうちどちらか一方を選択しメモリA～D（13～16）に入力し、マルチプレクサ22は、メモリA～D（13～16）の出力のうち一つを選択して出力する。

【0151】また、制御部23は、メモリA～D（13～16）、マルチプレクサ21、22の動作を制御する。

【0152】マルチプレクサ21は、記録時にはC1符号の符号器出力、再生時には再生データを選択するよう制御部により制御され、また、メモリA～D（13～16）の出力は、マルチプレクサ22により選択され、データ合成後のデータ、あるいは、データ分解後のデータとして出力される。

【0153】前記メモリA～D（13～16）は、それぞれ少なくとも符号語一つ分のデータが格納できるデータ容量を持つ。

【0154】図9に示す回路において、データ合成時の動作は前記図4に示す回路の動作の説明で用いた図5、図6のタイムチャートで表される。

【0155】一方、データ分解時の動作は、図7、図8のタイムチャートにおいてメモリE～H（17～20）

を、それぞれメモリA～D（13～16）と置き換えたものに等しい。

【0156】図9に示す回路によれば、図4に示す回路と同じ機能を、より少ないハードウェア量で実現することができる。

【0157】【実施例2】本実施例2は、異なる複数の積符号ブロックの符号語同士を合成するようにしたものであり、誤り訂正符号化回路、および、誤り訂正復号化回路の構成は、前記実施例と同じである。

【0158】図10は、本実施例2におけるデータ合成前の積符号ブロックのデータ配置を示す図である。

【0159】図10に示す左のブロックを積符号ブロック1、右のブロックを積符号ブロック2とする。

【0160】ただし、積符号ブロック1の方が積符号ブロック2よりも時間的には前に符号化されているとする。

【0161】これらの積符号ブロック（1、2）は、C1符号が（9、5）RS符号、C2符号が（6、4）RS符号から成る積符号ブロックであるとする。

【0162】また、各積符号ブロック（1、2）を構成するシンボルには、前記図2に示す積符号ブロックと同様に番号を付してある。

【0163】なお、二つの積符号ブロック（1、2）のデータを区別するため積符号ブロック1、積符号ブロック2のシンボル番号の先頭にはそれぞれA、Bを付してある。

【0164】図11は、本実施例2におけるデータ合成後の積符号ブロックのデータ配置を示す図である。

【0165】以下、二つの積符号ブロック（1、2）を合成し、共に積符号ブロックと同じサイズの二つの合成ブロックを生成する手順を、図10、図11を用いて説明する。

【0166】まず、図10に示す積符号ブロック1と積符号ブロック2のそれぞれ1段目の符号語（A1、A2、A3、A4、A5、A6、A7、A8、A9）、（B1、B2、B3、B4、B5、B6、B7、B8、B9）から二つの合成語を生成する。

【0167】この合成方法として、例えば、前記実施例1で示した方法を用いる場合には、二つの合成語として、（A1、B1、A2、B2、A3、B3、A4、B4、A5）、（B5、A6、B6、A7、B7、A8、B8、A9、B9）が生成される。

【0168】これら二つの合成語は、図11に示すように合成ブロック1の上2段に連続して配置される。

【0169】以下、積符号ブロック1、2の2段目、3段目についても同様に合成され合成ブロック1の配列上に配置される。

【0170】一方、積符号ブロック1、2の4段目から6段目については合成後、合成ブロック2の配列上に配置される。

【0171】このように生成された二つの合成ブロックは合成ブロック1、合成ブロック2の順で記録媒体に記録される。

【0172】図12は、本実施例2におけるデータ分解前のデータの誤り分布を示す図であり、図13は、本実施例2におけるデータ分解後のデータの誤り分布を示す図である。

【0173】次に、図12、図13を用いて、本実施例2におけるデータ合成の効果について説明する。

【0174】例えば、記録媒体から再生して得られた合成ブロックに、図12に示すような誤りが生じている場合について考える。

【0175】合成ブロック1では、上から2段目に4シンボル、3段目に3シンボル、5段目に3シンボルの誤りがそれぞれ発生している。

【0176】また、合成ブロック2では36シンボルのバースト誤りが発生している。

【0177】比較のため、まず、図12に示した分布の誤りがデータ合成を行わない場合に発生した場合について考える。

【0178】まず、合成ブロック1では、C1復号で上から2、3、5段目の誤りが訂正できず誤り検出フラグが付加される。

【0179】また、C2復号では、誤り検出フラグの数が2を超えているため、誤り検出フラグを用いないランダム訂正を行う。

【0180】その結果、左から2、3、6列目の誤りは訂正され、4、5、7列目にある7シンボルの誤りが残ることになる。

【0181】また、合成ブロック2に生じた誤りは、C1符号、C2符号の2段階の誤り訂正復号によっても誤りを訂正することができない。

【0182】即ち、データ合成を行わなかった場合には、二つのブロックで計43(= (7+36))シンボルの誤りが残ることになる。

【0183】これに対し、データ合成後の合成ブロックに対し図12の分布の誤りが生じた場合には、積符号復号の前にデータ分解を行う。

【0184】この結果図12の誤り分布は図13に示す誤り分布に変換される。

【0185】この状態で復号を行うと、積符号ブロック1については、まず、C1符号復号で上から1、2、3段目の誤りが訂正され、5、6段目に誤り検出フラグが付加される。

【0186】次に、C2符号復号では、C1復号時に付加された誤り検出フラグを用いてイレージャ訂正を行うことにより、上から5、6段目の誤りが全て訂正される。

【0187】積符号ブロック2については、C1復号で上から1、2、3段目の誤りが訂正され、5、6段目に

誤り検出フラグが付加される。

【0188】次に、C2復号ではC1復号時に付加された誤り検出フラグを用いてイレージャ訂正を行うことにより上から5、6段目の誤りが全て訂正され、図12の分布の誤りを全て訂正できる。

【0189】即ち、データ合成を行った場合には、比較的短いバースト誤りと長いバースト誤りの両方を効率良く訂正することができ、データ合成を行わない場合に比べ多くの誤りを訂正できる。

【0190】なお、前記説明では、理解しやすくするため、データを並べ替えることによって全ての誤りを訂正することが可能な場合について説明したが、誤りの種類によっては、データを並べ替えても全ての誤りを訂正することは不可能な場合があるが、そのような場合でも、訂正能力が向上することには変わりはない。

【0191】また、本実施例2では、異なる二つの積符号ブロックのデータを用いてデータ合成を行う場合について説明したが、データ合成を行う積符号ブロックの数は必ずしも二つである必要はなく三つ以上であってもよい。

【0192】また、必ずしも異なる積符号ブロックの各々同じ段の符号語同士を合成する必要はなく異なる段の符号語同士を合成してもよい。

【0193】また、データ合成の際のデータの並べ替えは、必ずしも1シンボル単位で行う必要はなく2シンボル以上を一単位として並べ替えてもよい。

【0194】さらに、前記実施例1と同様、データの並べ替えの規則はただ一通りではなく、連続するC1符号の符号長 $n-1$ より少ない個数のシンボルを一単位とした並べ替えを全て含むものとする。

【0195】図14は、本実施例2におけるデータ合成手段・データ分解手段の回路構成の一例を示す回路図である。

【0196】図14に示す回路は、メモリA~H(13~20)が積符号ブロック1個分のデータ容量を有する点以外は、前記図4に示す回路と同じである。

【0197】前記図10、図11に示すデータ合成を例に挙げて、図14に示すデータ合成部8・データ分解部9の動作について説明する。

【0198】まず、データ合成部8の動作について説明する。

【0199】図15は、図14に示すデータ合成部8の動作を示すタイムチャートである。

【0200】また、図16は、図15に示すタイムチャートの部分拡大図である。

【0201】ただし、「W」、「R」は、前記図5、12のタイムチャートと同様、それぞれメモリに対するデータの書き込み、メモリからのデータ読み出しを表している。

【0202】データ合成部8に対して積符号ブロック

1、積符号ブロック 2 の順で入力を与えられたとき、積符号ブロック 1 の前半の 27 シンボルは、メモリ A 13 の奇数アドレスに、後半の 27 シンボルは、メモリ B 14 の奇数アドレスに対してそれぞれ書き込まれる。

【0203】積符号ブロック 2 の前半の 27 シンボルは、メモリ A 13 の偶数アドレスに、後半の 27 シンボルは、メモリ B 14 の偶数アドレスに対してそれぞれ書き込まれる。

【0204】積符号ブロック 2 の全てのシンボルがメモリに書き込まれた時点で、メモリ A 13 には合成ブロック 1、メモリ B 14 には合成ブロック 2 が格納されている。

【0205】図 17 は、図 14 に示すデータ分解部 9 の動作を示すタイムチャートである。

【0206】また、図 18 は、図 17 に示すタイムチャートの部分拡大図である。

【0207】なお、図中「W'」とあるのは、図 18 に示すようにメモリに対する書き込みの実行を 1 シンボルおきに行うことを示している。

【0208】データ分解部 9 に対して入力データが合成ブロック 1、2 の順で与えられるとき、合成ブロック 1 の奇数番目のシンボル、偶数番目のシンボルは、それぞれメモリ E 17、メモリ F 18 の前半の連続するアドレスに対し書き込まれる。

【0209】合成ブロック 2 については、合成ブロック 1 とは逆に、偶数番目のシンボルがメモリ E 17 に、奇数番目のシンボルがメモリ F 18 に書き込まれる。

【0210】合成ブロック 2 のシンボルは、メモリ E 17、メモリ F 18 の後半のアドレスに、合成ブロック 1 のシンボルに引き続いて書き込まれる。

【0211】図 19 は、本実施例 2 におけるデータ合成手段・データ分解手段の回路構成の他の例を示す回路図であり、データ合成とデータ分解に用いるメモリを共用するようにした回路図である。

【0212】図 19 に示す回路は、メモリ A~D (13~16) が積符号ブロック 1 個分のデータ容量を有する点以外は、前記図 9 に示す回路と同じである。

【0213】図 19 に示す回路において、データ合成時の動作は前記図 14 に示す回路の動作の説明で用いた図 15、16 のタイムチャートで表される。

【0214】一方、データ分解時の動作は、図 17、18 のタイムチャートにおいてメモリ E~H (17~20) を、それぞれメモリ A~D (13~16) と置き換えたものに等しい。

【0215】図 19 に示す回路によれば、図 14 に示す回路と同じ機能を、より少ないハードウェア量で実現することができる。

【0216】〔実施例 3〕図 20 は、本発明の他の実施例（実施例 3）である誤り訂正符号化回路、および、誤り訂正復号化回路を用いたデジタル記録装置の概略構

成を示す図である。

【0217】本実施例 3 は、データ合成で生成された合成語にその合成語に関する ID 情報を付加する ID 情報付加手段 24 と、データ分解前に再生データの ID 情報を検査する ID 情報検査手段 25 とを、前記実施例 1 に付加したものである。

【0218】例えば、N 個（N は 2 以上の整数）の符号語を合成する場合には、ID 情報として、（1）積符号ブロックの配列上、どの組み合わせを用いて合成したのか、（2）合成して得られた N 個の合成語のうち何番目の符号語か、を示す情報を用いる。

【0219】図 21 は、前記実施例 1 の説明で用いた図 2 のデータ合成の場合の、本実施例 3 における ID 情報の一例を示す図である。

【0220】前記図 2 の例では、積符号ブロック内の連続する二つの符号語を合成して二つの合成語を生成する。

【0221】1 積符号ブロック当たり C 1 の符号語は 6 個あるから、符号語の組み合わせは、符号語 1 と 2、符号語 3 と 4、符号語 4 と 5 の 3 通りである。

【0222】従って、ID 情報のうち、前記（1）はこの 3 通りのうちどれかを示す値となり、2 ビットで表現できる。

【0223】また、（2）については（1）で示された符号語の組み合わせで生成される二つの合成語のうちのどちらかを示す値となり、1 ビットで表現できる。

【0224】従って、図 2 のデータ合成の場合には（1）、（2）を合わせて 3 ビットの ID 情報を各合成語に付加することになる。

【0225】一方、再生時にはデータを分解する前に合成語に付加された ID 情報を検査し、その結果に基づきデータをメモリに書き込みデータ分解を行う。

【0226】また、通常 ID 情報は、合成語の先頭に付加されるので、本実施例 3 では、ID 情報に基づき再生データをメモリに書き込む。

【0227】そのため、同期外れ等により合成語の先頭が正しく見つけられない場合に誤ったアドレスにデータを書き込むことを防止することができる。

【0228】従って、本実施例 3 によれば ID 情報を用いない場合に比べ、より信頼性の高いデジタルデータ記録装置を実現できる。

【0229】また、ID 情報付加手段 24 により ID 情報を付加した後に、ID 情報を保護するための手段、例えば、ID 情報に対して誤り検出符号で符号化を行う手段等の手段を備えることも可能である。

【0230】ここで、誤り検出符号とは誤りの検出のみで訂正はできない符号のことである。

【0231】再生時には、ID 情報検査に先立ち ID 情報に対して誤り検出を行い、誤りがないと判定された場合には、ID 情報にしたがい、データをメモリに書き込

む。

【0232】誤りがあると判定された場合には、その合成語のデータをメモリに書き込まずに捨てるか、あるいは、記録媒体から再び同じデータを再生し直す。

【0233】また、誤り検出符号の代わりに、誤り訂正符号でID情報を符号化してもよい。

【0234】この場合、再生時にはまずID情報部の誤り訂正復号を行い、ID情報部に生じた誤りを訂正した後に、ID情報に従いデータ分解を行うことができる。

【0235】ID情報部に生じた誤りが多く、訂正できなかった場合には、前記した誤り検出符号の場合と同様、その合成語のデータをメモリに書き込まずに捨てるか、あるいは、記録媒体から再び同じデータを再生し直す。

【0236】前記のようにID情報を保護するための手段を備えた場合には、より信頼性の高いデータ合成、データ分解を行うことができる。

【0237】以上、本発明の実施例についてデジタル記録装置を例に挙げて説明したが、本発明は、図24に示すようなデジタル通信装置に適用可能であることはいうまでもない。

【0238】以上、本発明を実施例に基づき具体的に説明したが、本発明は、前記実施例に限定されるものではなく、その要旨を逸脱しない範囲で種々変更し得ることは言うまでもない。

【0239】

【発明の効果】本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば、下記の通りである。

【0240】(1) 本発明によれば、C2符号、C1符号により2段階の符号化を行う積符号において、誤り訂正符号の仕様(符号長、パリティ数)を変えことなくC1符号の訂正能力を向上させることが可能となる。

【0241】(2) 本発明によれば、誤り率が低く信頼性の高いデジタル通信装置、または、デジタル記録装置を実現することが可能となる。

【図面の簡単な説明】

【図1】本発明の一実施例(実施例1)である誤り訂正符号化回路、および、誤り訂正復号化回路を用いたデジタル記録装置の概略構成を示す図である。

【図2】前記データ合成手段、データ分解手段の動作を説明するための図である。

【図3】図2示すデータ合成及びデータ分解の効果について説明するための図である。

【図4】本実施例1におけるデータ合成手段・データ分解手段の回路構成の一例を示す回路図である。

【図5】図4に示すデータ合成部8の動作を示すタイムチャートである。

【図6】図5に示すタイムチャートの部分拡大図である。

【図7】図4に示すデータ分解部9の動作を示すタイムチャートである。

【図8】図7に示すタイムチャートの部分拡大図である。

【図9】本実施例1におけるデータ合成手段・データ分解手段の回路構成の他の例を示す回路図である。

【図10】本実施例2におけるデータ合成前の積符号ブロックのデータ配置を示す図である。

【図11】本実施例2におけるデータ合成後の積符号ブロックのデータ配置を示す図である。

【図12】本実施例2におけるデータ分解前のデータの誤り分布を示す図である。

【図13】本実施例2におけるデータ分解後のデータの誤り分布を示す図である。

【図14】本実施例2におけるデータ合成手段・データ分解手段の回路構成の一例を示す回路図である。

【図15】図14に示すデータ合成部8の動作を示すタイムチャートである。

【図16】図15に示すタイムチャートの部分拡大図である。

【図17】図14に示すデータ分解部9の動作を示すタイムチャートである。

【図18】図17に示すタイムチャートの部分拡大図である。

【図19】本実施例2におけるデータ合成手段・データ分解手段の回路構成の他の例を示す回路図である。

【図20】図20は、本発明の他の実施例(実施例3)である誤り訂正符号化回路、および、誤り訂正復号化回路を用いたデジタル記録装置の概略構成を示す図である。

【図21】前記実施例1の説明で用いた図2のデータ合成の場合の、本実施例3におけるID情報の一例を示す図である。

【図22】積符号の符号化方法を説明するための図である。

【図23】積符号ブロックの構成は変更せずに、より誤り訂正能力を向上させる従来方法を説明するための図である。

【図24】デジタル通信システムにおいて使用される、無線を使用するデジタル通信装置の概略構成を示すブロック図である。

【符号の説明】

1…C2符号符号化手段、2…C1符号符号化手段、3…データ合成手段、4…記録・再生系、5…C2符号復号化手段、6…C1符号復号化手段、7…データ分解手段、8…データ合成部、9…データ分解部、10、23…制御部、11、12、21、22…マルチプレクサ、13～20…メモリ、24…ID情報付加手段、25…ID情報検査手段、31…データ配置手段、32…データ送出手段、33…積符号生成手段、34…積符号復号

手段。

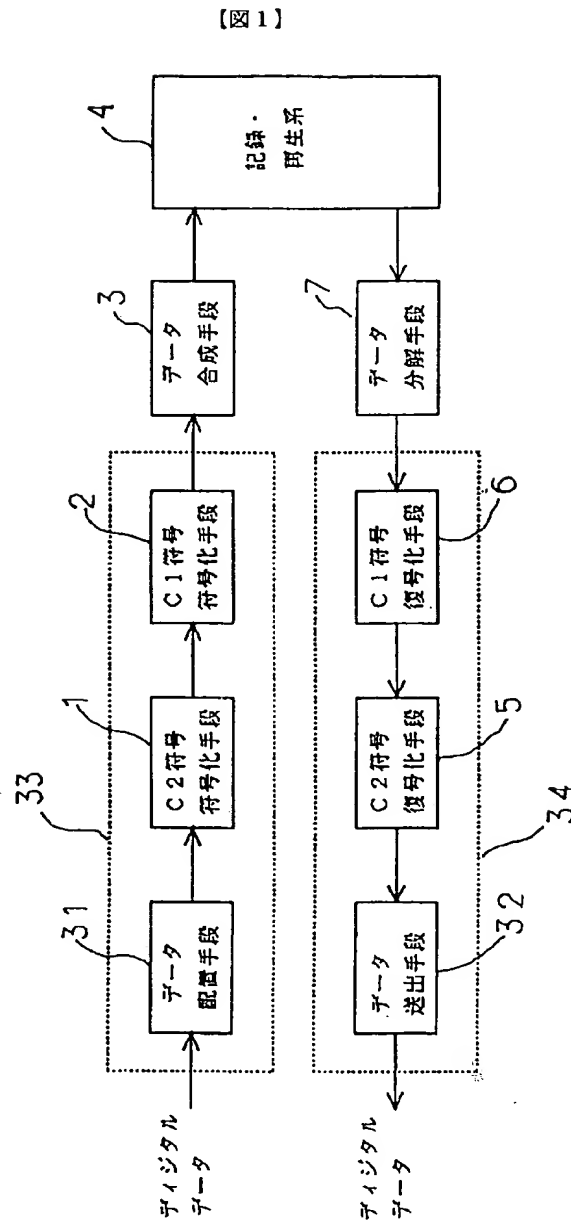
23

(13)

特開平8-125548

24

図1



【図2】

図 2

積符号ブロック									
A1	A2	A3	A4	A5	A6	A7	A8	A9	
A1	A2	A3	A4	A5	A6	A7	A8	A9	符号語 1
A10	A11	A12	A13	A14	A15	A16	A17	A18	符号語 2
A19	A20	A21	A22	A23	A24	A25	A26	A27	符号語 3
A28	A29	A30	A31	A32	A33	A34	A35	A36	符号語 4
A37	A38	A39	A40	A41	A42	A43	A44	A45	符号語 5
A46	A47	A48	A49	A50	A51	A52	A53	A54	符号語 6

データ合成 ↓ ↑ データ分解
合成ブロック

A1	A10	A2	A11	A3	A12	A4	A13	A5	
A1	A10	A2	A11	A3	A12	A4	A13	A5	
A14	A6	A15	A7	A16	A8	A17	A9	A18	合成語 1
A19	A28	A20	A29	A21	A30	A22	A31	A23	合成語 2
A32	A24	A33	A25	A34	A26	A35	A27	A36	合成語 3
A37	A46	A38	A47	A39	A48	A40	A49	A41	合成語 4
A50	A42	A51	A43	A52	A44	A53	A45	A54	合成語 5
									合成語 6

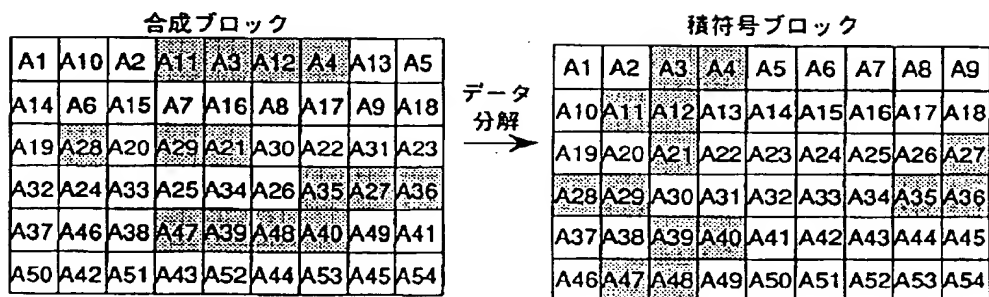
【図21】

図 2 1

合成語 \ ID情報	(1)		(2)
合成語 1	0	0	0
合成語 2	0	0	1
合成語 3	0	1	0
合成語 4	0	1	1
合成語 5	1	0	0
合成語 6	1	0	1

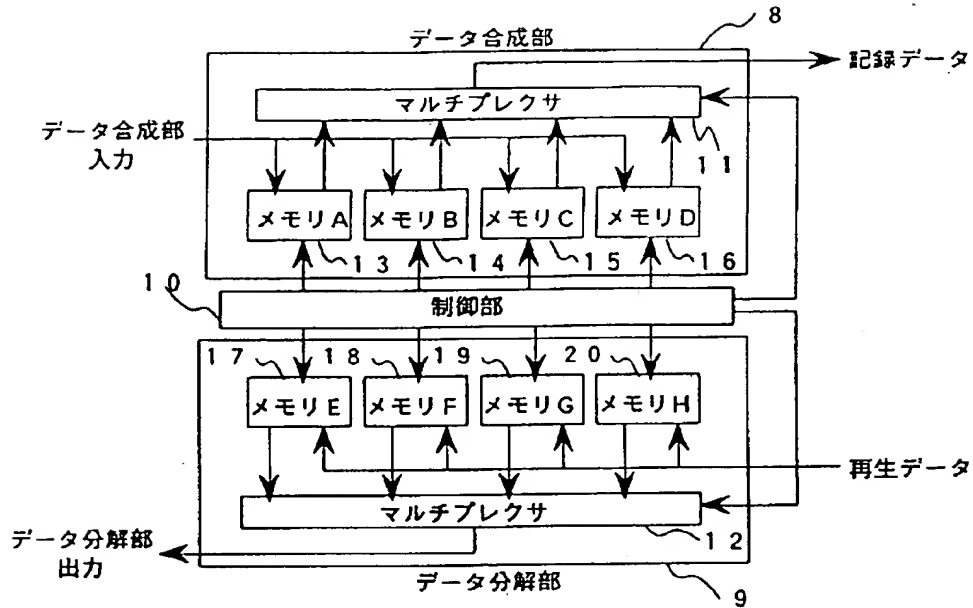
【図3】

図 3



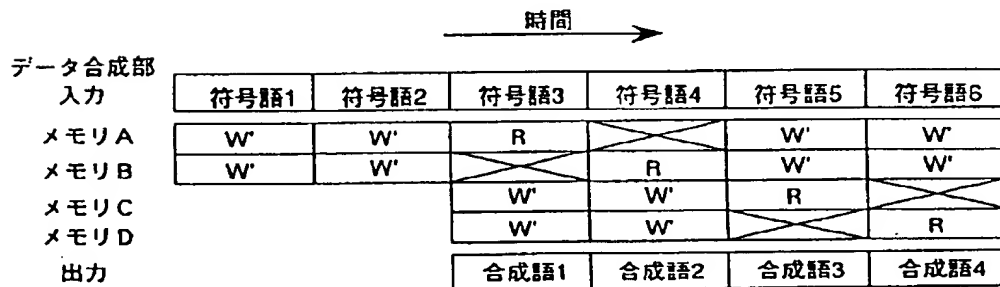
【図4】

図4



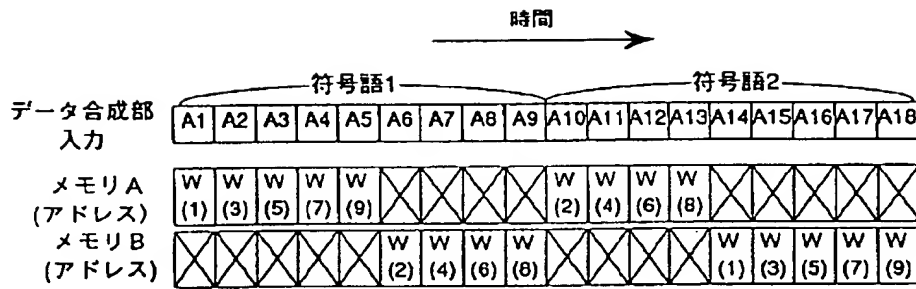
【図5】

図5



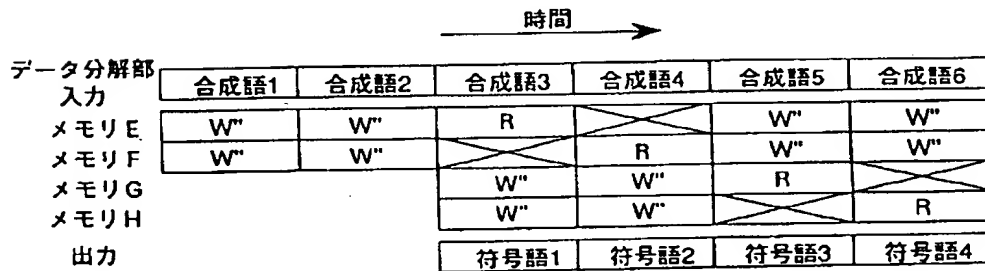
【図 6】

図 6



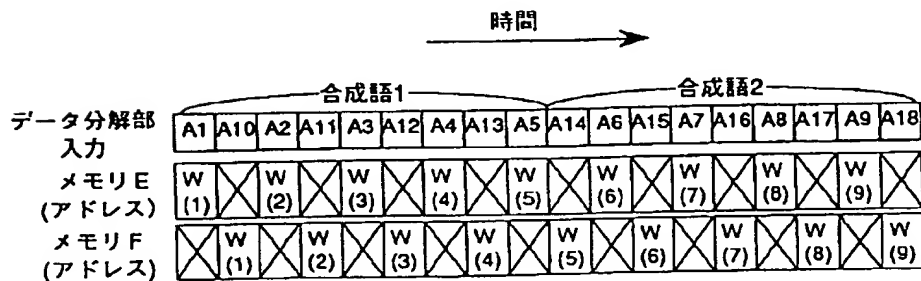
【図 7】

図 7



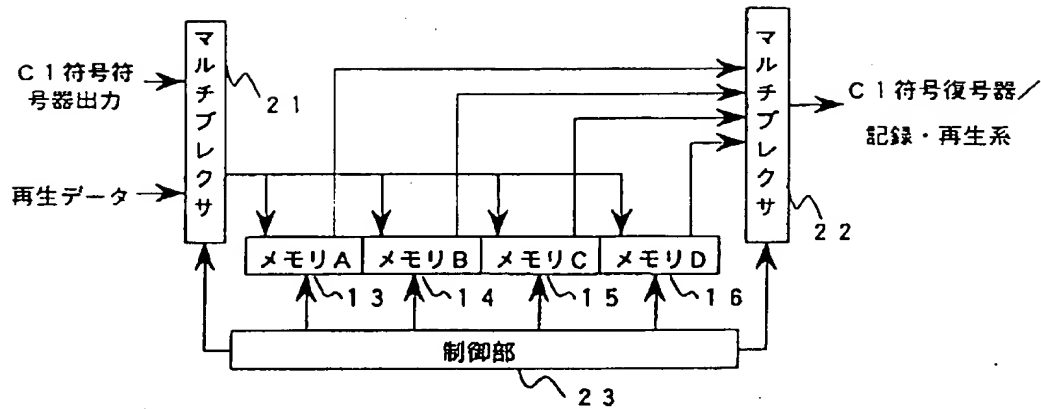
【図 8】

図 8



【図 9】

図 9



【図 10】

図 10

積符号ブロック 1										積符号ブロック 2									
A1	A2	A3	A4	A5	A6	A7	A8	A9		B1	B2	B3	B4	B5	B6	B7	B8	B9	
A10	A11	A12	A13	A14	A15	A16	A17	A18		B10	B11	B12	B13	B14	B15	B16	B17	B18	
A19	A20	A21	A22	A23	A24	A25	A26	A27		B19	B20	B21	B22	B23	B24	B25	B26	B27	
A28	A29	A30	A31	A32	A33	A34	A35	A36		B28	B29	B30	B31	B32	B33	B34	B35	B36	
A37	A38	A39	A40	A41	A42	A43	A44	A45		B37	B38	B39	B40	B41	B42	B43	B44	B45	
A46	A47	A48	A49	A50	A51	A52	A53	A54		B46	B47	B48	B49	B50	B51	B52	B53	B54	

【図 13】

図 13

積符号ブロック 1										積符号ブロック 2									
A1	A2	A3	A4	A5	A6	A7	A8	A9		B1	B2	B3	B4	B5	B6	B7	B8	B9	
A10	A11	A12	A13	A14	A15	A16	A17	A18		B10	B11	B12	B13	B14	B15	B16	B17	B18	
A19	A20	A21	A22	A23	A24	A25	A26	A27		B19	B20	B21	B22	B23	B24	B25	B26	B27	
A28	A29	A30	A31	A32	A33	A34	A35	A36		B28	B29	B30	B31	B32	B33	B34	B35	B36	
A37	A38	A39	A40	A41	A42	A43	A44	A45		B37	B38	B39	B40	B41	B42	B43	B44	B45	
A46	A47	A48	A49	A50	A51	A52	A53	A54		B46	B47	B48	B49	B50	B51	B52	B53	B54	

【図 11】

図 1 1

合成ブロック 1										合成ブロック 2									
A1	B1	A2	B2	A3	B3	A4	B4	A5		A28	B28	A29	B29	A30	B30	A31	B31	A32	
B5	A6	B6	A7	B7	A8	B8	A9	B9		B32	A33	B33	A34	B34	A35	B35	A36	B36	
A10	B10	A11	B11	A12	B12	A13	B13	A14		A37	B37	A38	B38	A39	B39	A40	B40	A41	
B14	A15	B15	A16	B16	A17	B17	A18	B18		B41	A42	B42	A43	B43	A44	B44	A45	B45	
A19	B19	A20	B20	A21	B21	A22	B22	A23		A46	B46	A47	B47	A48	B48	A49	B49	A50	
B23	A24	B24	A25	B25	A26	B26	A27	B27		B50	A51	B51	A52	B52	A53	B53	A54	B54	

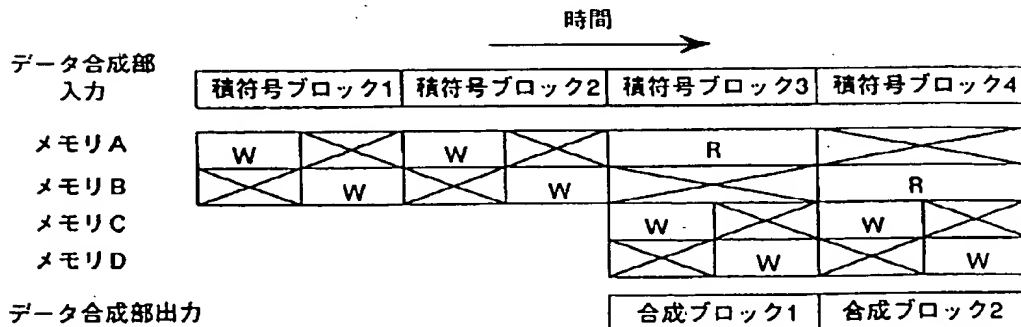
【図 12】

図 1 2

合成ブロック 1										合成ブロック 2									
A1	B1	A2	B2	A3	B3	A4	B4	A5		A28	B28	A29	B29	A30	B30	A31	B31	A32	
B5	A6	B6	A7	B7	A8	B8	A9	B9		B32	A33	B33	A34	B34	A35	B35	A36	B36	
A10	B10	A11	B11	A12	B12	A13	B13	A14		A37	B37	A38	B38	A39	B39	A40	B40	A41	
B14	A15	B15	A16	B16	A17	B17	A18	B18		B41	A42	B42	A43	B43	A44	B44	A45	B45	
A19	B19	A20	B20	A21	B21	A22	B22	A23		A46	B46	A47	B47	A48	B48	A49	B49	A50	
B23	A24	B24	A25	B25	A26	B26	A27	B27		B50	A51	B51	A52	B52	A53	B53	A54	B54	

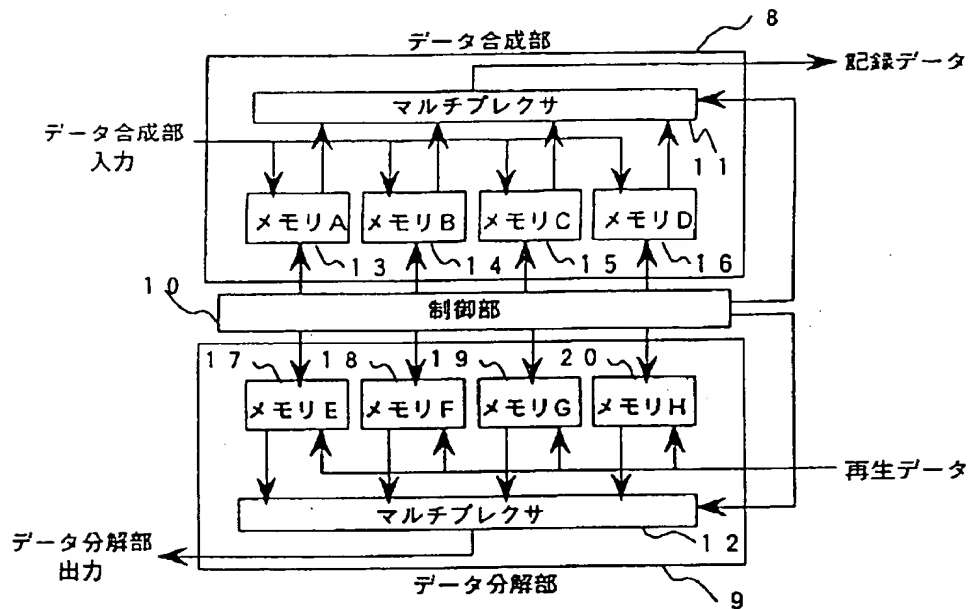
【図 15】

図 1 5



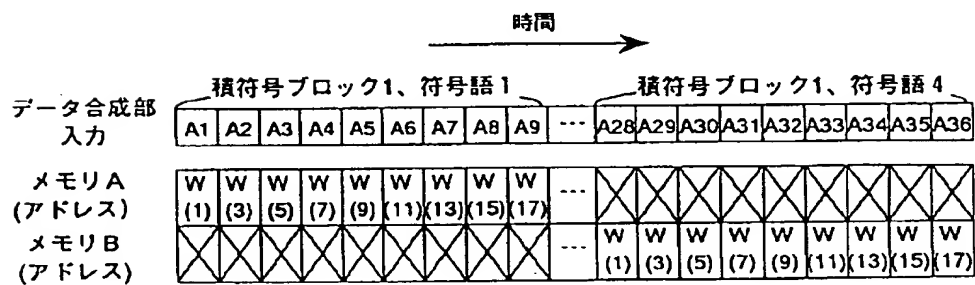
【図14】

図 14



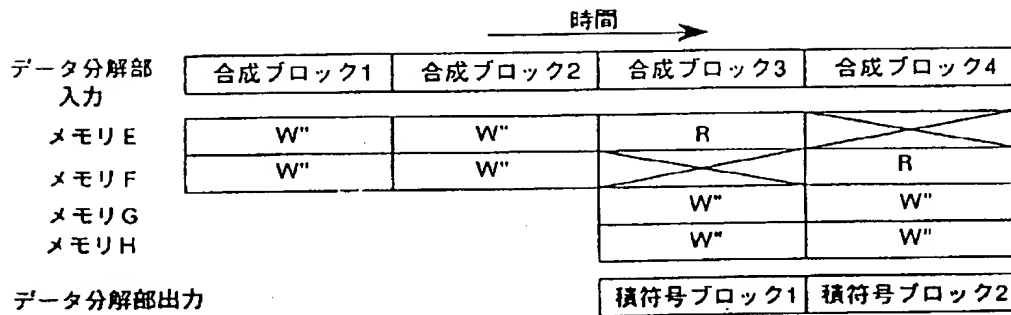
【図16】

図 16



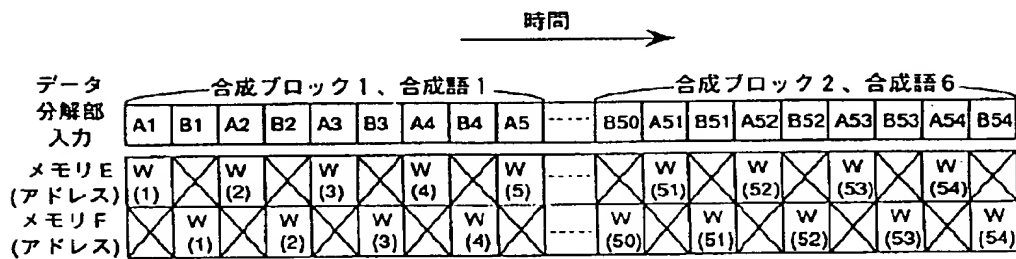
【図17】

図 1 7



【図18】

図 1 8



【図19】

図 1 9

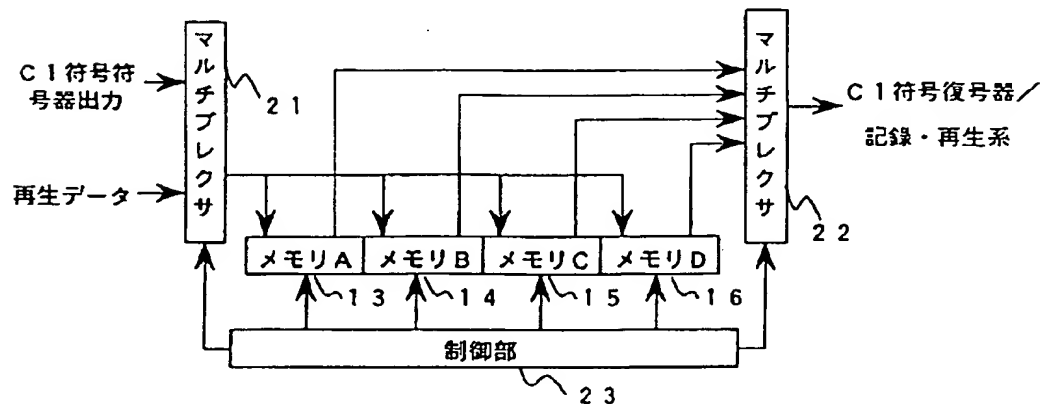


図 20

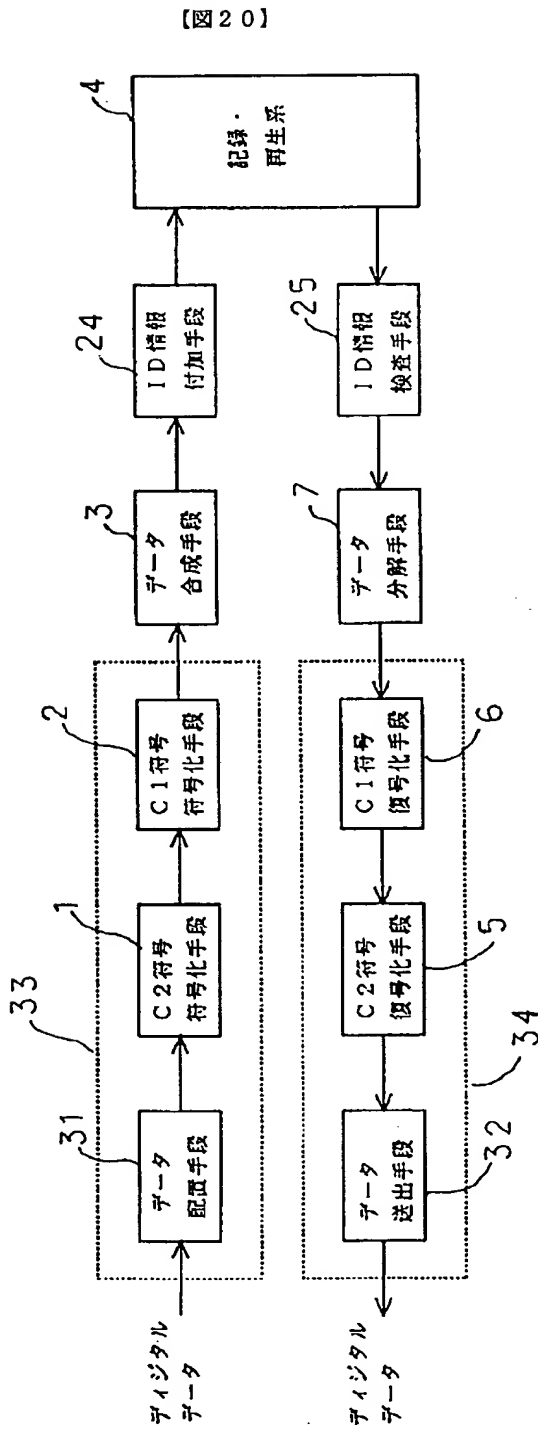
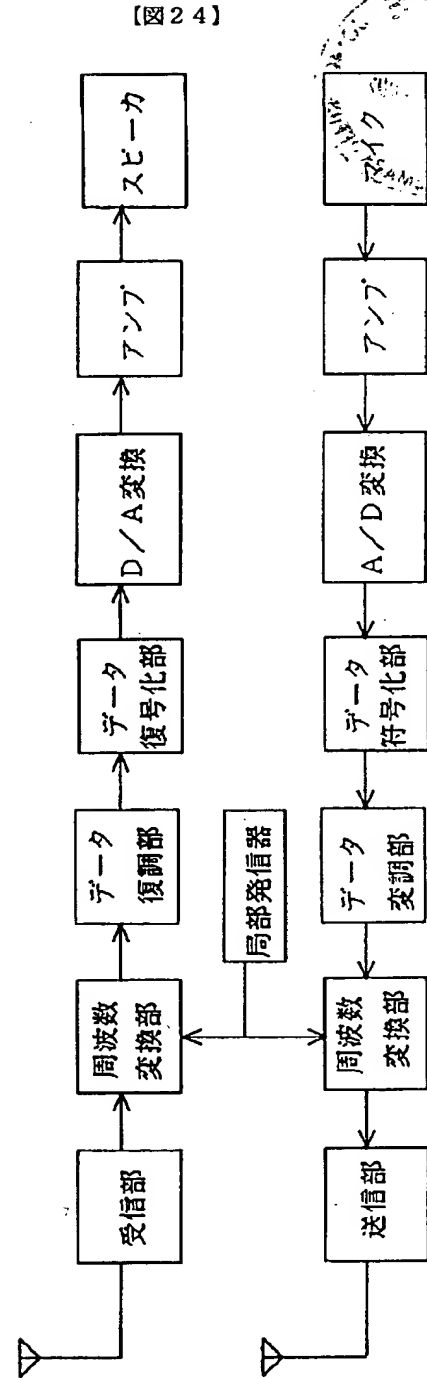
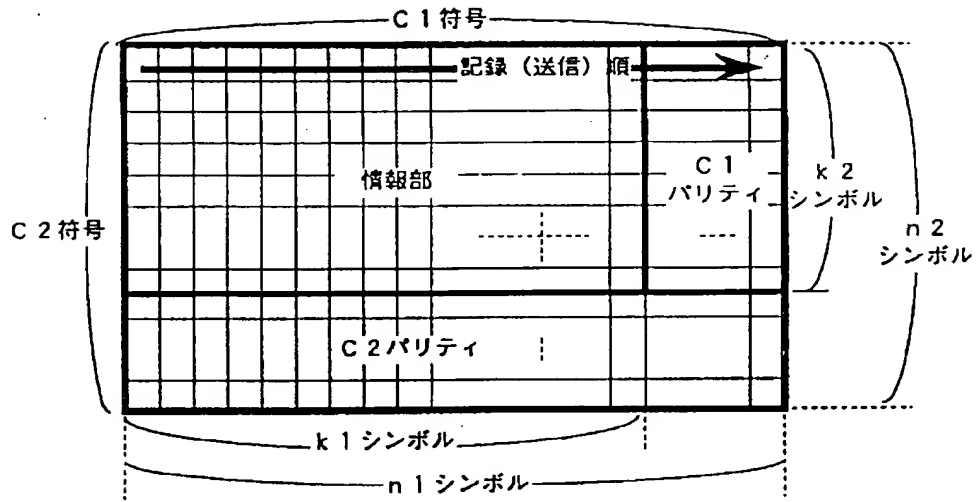


図 24



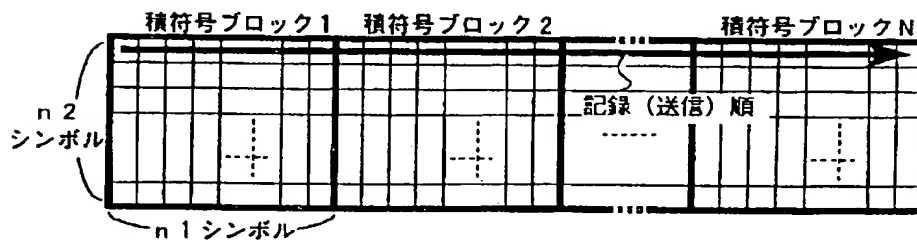
【図22】

図 2 2



【図23】

図 2 3





English Translation of the Specification, JP-A-8-125548; page 3, right column, line 37 to page 5, left column, line 28

[0017] Fig. 22 is a diagram to explain a method of coding a product code.

[0018] Description will be next given of the method of coding a product code by referring to Fig. 22.

[0019] In this example of a product code, a C1 code is represented as an (n_1, k_1) RS code and a C2 code is represented as an (n_2, k_2) RS code.

[0020] In this expression, an (n, k) RS code indicates a Reed-Solomon code, where n is the code length and k is the number of information symbols.

[0021] First, information to be recorded is arranged in a 2-dimensional array, a k_1 symbols \times k_2 symbols array as shown in Fig. 22.

[0022] In Fig. 22, each grid corresponds to one symbol.

[0023] Next, an error correction code is coded or generated for the C2 code in the vertical direction to add a parity including $(n_2 - k_2)$ symbols to resultantly generate a code word of an (n_2, k_2) RS code.

[0024] Next, an error correction code is generated for the C1 code in the horizontal direction to add a parity including $(n_1 - k_1)$ symbols to resultantly generate a code word of (n_1, k_1) RS code.

[0025] The code including $(n_1 \times n_2)$ symbols obtained by the two-stage coding of the C1 code and the C2 code is a product code.

[0026] The 2-dimensional block including $(n_1 \times n_2)$ symbols will be referred to as a product code block hereinbelow in this specification.

[0027] Data of the product code block obtained by the coding described above is ordinarily recorded on a magnetic tape in the unit of the code word of C1 code starting at the highest row in a direction indicated by an arrow mark.

[0028] On the other hand, when reproducing the data, the data sequentially reproduced from the magnetic tape is arranged in a 2-dimensional array in the same data sequence as the data recording sequence, and the error correction code is decoded for the C1 code and the C2 code in this order.

[0029] Since the parity number of the C1 code is $(n_1 - k_1)$ symbols, the maximum number of correctable symbols is $(n_1 - k_1)/2$ symbols.

[0030] An error detection flag is set to all symbols of the code word which cannot be corrected using the decoding of the error correction code for the C1 code.

[0031] In the decoding of the error correction code for the C2 code, when the number of error detection flags added in the decoding of the C1 code is equal to or less than $(n_2 - k_2)$, an erasure correction is conducted assuming that the position of each error detection flag is an error position.

[0032] Since the parity number of the C2 code is $(n_2 - k_2)$, symbols up to the number of $(n_2 - k_2)/2$ may be corrected with the erasure correction.

[0033] On the magnetic tape, consecutive $((n_2 - k_2) \times n_1)$ symbols of errors are $(n_2 - k_2)$ symbols of errors in the C2 code and are therefore correctable.

[0034] That is, in the product code, $((n_2 - k_2) \times n_1)$ symbols of burst errors on the magnetic tape can be corrected.

[0035] When the number of error detection flags exceeds $(n_2 - k_2)$, a random correction is conducted as in the case of the C1 code.

[0036] There has been heretofore known a method of coding the product code which improves the error correction efficiency without changing the block layout of the product code.

[0037] A product code coding method of the prior art which improves the error correction efficiency without changing the block layout of the product code will now be described.

[0038] As shown in Fig. 23, N product code blocks are arranged in the horizontal direction so that data is recorded on a magnetic tape beginning at the highest column in a direction of an arrow mark, where N is an integer equal to or more than two.

[0039] Resultantly, data of code words of the C1 code of different product code blocks is sequentially recorded on the tape.

[0040] This is equivalent to an operation to rearrange data in the unit of code word of the C1 code for different product code blocks.

[0041] The data rearrangement distributes the successive burst errors on the magnetic tape into N product-code blocks. For each product-code block, the burst error length becomes virtually $1/N$ of the original length.

[0042] For example, burst errors having a length of $(n1 \times N)$ symbols on a magnetic tape become $n1$ symbols of burst errors in each product-code block.

[0043] Therefore, the length of correctable burst errors is $((n2 - k2) \times n1 \times N)$ symbols. This is N times that of the operation in which data of the respective blocks are collectively recorded.

[0044] That is, the rearrangement of data in the unit of code word makes it possible to efficiently use the correction performance of the product code to correct the burst errors.

[0045]

[Problem to be solved by the Invention] In the data rearranging method of the prior art, data is rearranged in the code unit of C1 code. This rearrangement can disperse burst errors having a length exceeding the code length of C1 code into a plurality of product-code blocks.

[0046] However, since the code words are collectively arranged, burst errors having a length equal to or less than the code length of C1 code cannot be dispersed.

[0047] Consequently, it is not necessarily possible for the method of the prior art to fully use the error correction

performance of the error correction code, and a target error ratio cannot be obtained in some cases.

[0048] It is therefore an object of the present invention, which has been devised to solve the problem of the prior art, to provide a technique, for use with an error correction code coding circuit and an error correction code decoding circuit, to efficiently conduct an error correction for burst errors having a length less than the code length of C1 code.

[0049] The object, other objects, and novel features of the present invention will become more apparent from the consideration of the following detailed description taken in conjunction with the accompanying drawings.

[0050]

[Means for Solving Problem] Description will now be briefly given of an outline of representative one of the invention disclosed by the present application.

[0051] (1) The present invention is characterized by comprising product-code generating means which handles information data arranged in a 2-dimensional ($k_1 \times k_2$) array such that the means codes an error correction code for each set of k_2 information symbols of each column of the 2-dimensional array using an (n_2, k_2) C2 code, where n_2 is a code length, codes an error correction code for each set of k_1 information symbols of each row of the 2-dimensional array using an (n_1, k_1) C1 code, where n_1 is a code length, and resultantly generates a product code including $(n_1 \times n_2)$ symbols and data rearranging means which rearranges data using, as one unit, K successive symbols of N code words

selected from a plurality of code words of C1 code included in the $(n1 \times n2)$ -symbol product code, where N is an integer equal to or more than two and $1 \leq K \leq n1$, and resultantly generates N new data sequences, each sequence including $n1$ symbols.

[0052] (2) The present invention is characterized in that in the means of (1) above, the N code words are included in at least two different product codes.

[0053] (3) The present invention is characterized in that the means of (1) or (2) above further includes an identifier (ID) information adding means for adding information regarding the data arrangement of the new data sequences.

[0054] (4) The present invention is characterized in that in the means of (1), (2), or (3) above, the data rearranging means includes at least a plurality of memories and a controller to control read and write operations of the memories. Under control of the controller, the data rearranging means selects K consecutive symbols from N code words of C1 code included in the product code including $(n1 \times n2)$ symbols, writes the selected K symbols in a data rearranging order in the memories, and then reads the data from the memories to resultantly generate N new data sequences each of which includes $n1$ symbols, where N is an integer equal to or more than two and $1 \leq K \leq n1$.

FIG. 22 (図22)

図 2 2

